

STEPPING ＆ SERVO MOTOR CONTROLLER'S OPTION

MPL-28-01V1.00/USBW32

取扱説明書
（設計者用）

（デバイスドライバ AL I/O ユニット編）

CB−23/USB
CB−08
CB−34/IO

USER'S MANUAL

本製品を使用する前に、この取扱説明書を良く読んで
十分に理解してください。
この取扱説明書は、いつでも取り出して読めるように
保管してください。

MN0145

目次

1. 概要

MPL-28-01V1.00/USBW32 は、DOS/V パソコンの Windows 上で CB-23/USB（AL シリーズ対応 USB I/F ユニット）を使用し、AL シリーズ(弊社オリジナルステッピング＆サーボモータコントローラシステム）上のスレーブユニットを動作させるための DLL ベースの関数ライブラリです。
各関数は、次に示すスレーブユニット上の PORT のアクセス(読み出し/書き込み)を行う為のものです。スレーブユニット上の PORT については、各スレーブユニットの取扱説明書を御覧ください。
◎ C-770AL
◎ CB-08
◎ CDB-5420-AL770
◎ CAD-5410-AL770
◎ C-771
◎ CD-773/ADB5331A
◎ C-772
◎ CB-34/IO

以降、MPL-28-01V1.00/USBW32 は、MPL-28 と称します。

2. 取扱説明書の構成

MPL-28 は、次に示すようにユニットタイプで分類された取扱説明書が用意されていますので、ご使用になるユニットに該当する取扱説明書をご覧ください。

ユニット	ユニット	取扱説明書
I/F ユニット	パソコン-スレーブユニット間のインタフェースユニット ***CB-23/USB***	注.
MCC05 ユニット	パルスジェネレータ MCC05 を搭載したスレーブユニット ***C-770AL*** ***CDB-5420-AL770*** ***CAD-5410-AL770*** ***C-771*** ***CD-773/ADB5331A***	***AL MCC05 ユニット編***
MCC06 ユニット	パルスジェネレータ MCC06 を搭載したスレーブユニット ***C-772***	***AL MCC06 ユニット編***
I/O ユニット	汎用 I/O ポートを持つスレーブユニット ***CB-08*** ***CB-34/IO***	***AL I/O ユニット編*** （当取扱説明書）

注.AL MCC05 ユニット編、AL MCC06 ユニット編、AL I/O ユニット編いずれでも示しています。

3. ご使用になる前に

3-1. DLL の使用上の注意

MPL-28 は、次に示すようにユニットタイプで分類された 4 つの DLL で構成されています。
ユーザアプリケーションは、使用するユニットに該当する DLL の関数を呼び出して、モータコントロール及び、汎用 I/O PORT へのアクセスを行います。よってユーザアプリケーションは、最低でも 2 つの DLL が必要になります。
例.C-772 使用の場合
AlUsbA.dll、Ac06A.dll が必要になります。
これによりヘッダーファイル(モジュール定義ファイル)、ライブラリーファイルも複数必要になります。

ユニット	ユニット	DLL
I/F ユニット	パソコン-スレーブユニット間のインタフェースユニット *CB-23/USB*	AlUsbA.dll
MCC05 ユニット	パルスジェネレータ MCC05 を搭載したスレーブユニット *C-770AL* *CDB-5420-AL770* *CAD-5410-AL770* *C-771* *CD-773/ADB5331A*	Ac05A.dll
MCC06 ユニット	パルスジェネレータ MCC06 を搭載したスレーブユニット *C-772*	Ac06A.dll
I/O ユニット	汎用 I/O ポートを持つスレーブユニット *CB-08* *CB-34/IO*	AcIoA.dll

3-2. この取扱説明書の見方

(1) 各種 DLL の関数と構造体

この取扱説明書の、各種 DLL の構造体、または関数は、次の形式で解説します。

○○○○構造体 ← 構造体の名称　　　構造体が使用される関数が存在する DLL の名称 → ○○○○ DLL

○○○○ ← 構造体が使用される関数が対応するマスターボードまたはスレーブユニットの名称

説　明

・・・・・・・・・・・・・・・・・・・・・・・・ → 構造体の説明

書　式

C 言語 ・・・・・・・・・・・・・・・・・・・・・・ → C 言語で、構造体を使用するときの定義

VB ・・・・・・・・・・・・・・・・・・・・・・ → Visual Basic で、構造体を使用するときの定義

VB.NET ・・・・・・・・・・・・・・・・・・・・・・ → Visual Basic.NET で、構造体を使用するときの定義

Delphi ・・・・・・・・・・・・・・・・・・・・・・ → Delphi で、構造体を使用するときの定義

メンバ

・・・・・・・・・・・・・・・・・・・・・・・・ → 構造体のメンバに格納される値の説明

○○○○関数 ← 関数の名称　　　関数が存在する DLL の名称 → ○○○○ DLL

○○○○ ← 関数が対応するマスターボードまたはスレーブユニットの名称

機　能

・・・・・・・・・・・・・・・・・・・・・・・・ → 関数の機能の説明

書　式

C 言語 ・・・・・・・・・・・・・・・・・・・・・・ → C 言語で、関数を使用するときの定義

VB ・・・・・・・・・・・・・・・・・・・・・・ → Visual Basic で、関数を使用するときの定義

VB.NET ・・・・・・・・・・・・・・・・・・・・・・ → Visual Basic.NET で、関数を使用するときの定義

Delphi ・・・・・・・・・・・・・・・・・・・・・・ → Delphi で、関数を使用するときの定義

引　数

・・・・・・・・・・・・・・・・・・・・・・・・ → 関数の各引数に指定する値の説明

戻り値

・・・・・・・・・・・・・・・・・・・・・・・・ → 関数の戻り値の説明

3-3.　マルチプロセス対応

複数のユーザアプリケーションで同時に当 DLL を使用することはできません。

3-4.　パソコンと I/F ユニット間の通信エラー

USB 通信 I/O エラーが発生した場合、環境設定ツール関数または環境設定直接関数を実行しない限り、エラーはクリアされず、USB 通信は行われません。これらのエラーが発生した場合、再度、環境設定ツール関数または環境設定直接関数を実行してください。

● USB 通信 I/O エラー
・リクエストの最終バイト送信後、200ms 経過しても、アンサーバックの先頭バイトを受信できない
・アンサーバックの受信中に、受信データが得られないまま 200ms 以上経過
・I/F ユニットから受信したデータに異常がある。

【注意】
リトライ回数が 0 以外に設定されいる場合、USB 通信エラー、USB タイムアウトエラー発生後、リトライを行いますが、Windows に割り込まれた場合、リトライ処理に遅れが生じます。

3-5.　AL 通信上の通信エラー

AL 通信上の通信エラーが発生した場合、環境設定ツール関数または環境設定直接関数を実行しない限り、エラーはクリアされません。AL 通信上の通信エラーが発生した場合、再度、環境設定ツール関数または環境設定直接関数を実行してください。

※ AL 通信エラーについては、CB-23/USB の取扱説明書をご覧ください。

4. AL I/F ユニット対応 DLL

4-1. 環境設定

AL I/F ユニット対応 DLL では、リトライ回数、AL 通信ボーレートを内部に記憶しています。これらの情報のことを環境設定情報と称します。MPL-28 の各種ライブラリ関数を実行することによる USB 通信、AL シリーズ通信は、環境設定情報をもとに行われるため、ユーザアプリケーションは、次のいずれかの方法により、環境設定情報を設定しなければなりません。

この章では、AL I/F ユニット対応 DLL を、単に I/F DLL と称します。

(1) 環境設定ツールによる設定

環境設定ツールは、画面上から手動操作により環境設定情報を設定するためのツールです。環境設定ツールを使用した場合、環境設定情報に変更があっても、ユーザアプリケーションの修正と再コンパイルが不要になります。環境設定ツールで設定した環境設定情報は、環境設定ツール関数で I/F DLL 内部に取り込まれます。この情報をもとに、USB **通信**のオープン、I/F ユニットの初期化が実行されるため、環境設定ツール関数は、ユーザアプリケーションの先頭で実行してください。

(2) ユーザアプリケーションによる直接設定

ユーザアプリケーションから環境設定直接関数を実行して、環境設定情報を指定します。環境設定直接関数が実行されると、指定された環境設定情報は、I/F DLL 内部に取り込まれます。この情報をもとに、USB 通信のオープン、I/F ユニットの初期化が実行されるため、環境設定直接関数は、ユーザアプリケーションの先頭で実行してください。

4-2. 構造体と関数

● **構造体一覧**

構造体名	説明
RESULT 構造体 環境設定ツール情報構造体 スレーブ情報構造体	関数を実行した結果を格納 環境設定情報を格納 全スレーブのスレーブタイプを格納

● **関数一覧**

関数名	機能
環境設定ツール関数 環境設定直接関数 環境設定ツール情報読み出し関数 環境設定問い合わせ関数 AL 通信エラー累計回数読み出し関数 AL 通信エラー累計回数クリア関数 USB 通信エラー累計回数読み出し関数 USB 通信エラー累計回数クリア関数 スレーブ情報読み出し関数	環境設定ツールで設定した情報で環境設定 設定値を直接指定して環境設定 環境設定ツールで設定された情報の読み出し すでに環境設定が行われているかの問い合わせ AL 通信上で発生したエラーの累計回数の読み出し AL 通信エラー累計回数のクリア USB 通信エラーの累計回数の読み出し USB 通信エラー累計回数のクリア 全スレーブのスレーブタイプの読み出し

RESULT 構造体

AL I/F ユニット対応 DLL

CB-23/USB

説　明

I/F DLL の関数を実行した結果が格納されます。

書　式

C 言語
```
typedef struct ALK_TAG_S_RESULT {
    WORD ALK_Result[4];
} ALK_S_RESULT;
```

VB
```
Type ALK_S_RESULT
    ALK_Result(1 To 4) As Integer
End Type
```

VB.NET
```
Structure ALK_S_RESULT
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=4)> Public ALK_Result() As Short
    Public Sub Initialize()
        ReDim ALK_Result(4)
    End Sub
End Structure
```

Delphi
```
ALK_S_RESULT = record
    ALK_Result: array[1..4] of WORD;
end;
```

メンバ

次に示すメンバは、C 言語で表記しています。C 言語の *ALK_Result[0]*～ *ALK_Result[3]*は、Visual Basic では *ALK_Result(1)*～ *ALK_Result(4)*、Visual Basic.NET では *ALK_Result(0)*～ *ALK_Result(3)*、Delphi では *ALK_Result[1]*～ *ALK_Result[4]*に対応します。

ALK_Result[0]　・・・ 実行された関数を示します。このメンバに格納される値は、次のいずれかです。

値	実行された関数	値	実行された関数
1	環境設定ツール関数	102	AL 通信エラー累計回数読み出し関数
2	環境設定直接関数	103	AL 通信エラー累計回数クリア関数
100	環境設定ツール情報読み出し関数	104	USB 通信エラー累計回数読み出し関数
101	スレーブ情報読み出し関数	105	USB 通信エラー累計回数クリア関数

値は 10 進表記です。

ALK_Result[1]　・・・ 実行結果を示します。このメンバに格納される値は、次のいずれかです。

値	実行結果
0	関数の実行が正常に終了しました
1	DLL 内部で API エラーが発生しました
2	NULL ポインタが指定されています
4	指定したパラメータが設定範囲を超えています
8	環境設定されていません
19	I/F ユニットが検出できません
20	USB 通信時に I/O エラーが発生しました
31	DLL が複数のプロセスから同時に使用されました

値は 10 進表記です。

ALK_Result[2] ・・・ I/F ユニット、スレーブユニット上で発生したエラーを示します。
このメンバに格納される値は、次のいずれかです。

値		実行結果
0	（H'00）	エラーは発生しませんでした
1 ～ 127	（H'01 ～ H'7F）	スレーブユニット固有のエラーです。各スレーブの取扱説明書をご覧ください
128	（H'80h）	初期化エラー（スレーブユニットが不正に電源 OFF または RESET されました）
129	（H'81h）	シリアルエラー（スレーブユニットからの受信時にエラーが発生しました）
130	（H'82h）	タイムアウトエラー（スレーブユニットへの送信時にエラーが発生しました）

値は 10 進表記です。()内は、16 進表記です。

ALK_Result[3] ・・・ 将来の拡張用です。

環境設定ツール情報構造体

AL I/F ユニット対応 DLL

CB-23/USB

説　明

環境設定情報が格納されます。

書　式

C 言語

```
typedef struct ALK_TAG_S_ENVIRONMENT_INFO {
    WORD ALK_Ch1;
    WORD ALK_Ch2;
    WORD ALK_Ch3;
    WORD ALK_Retry;
    WORD ALK_RsBaudRate;
    WORD ALK_AlBaudRate;
} ALK_S_ENVIRONMENT_INFO;
```

VB

```
Type ALK_S_ENVIRONMENT_INFO
    ALK_Ch1 As Integer
    ALK_Ch2 As Integer
    ALK_Ch3 As Integer
    ALK_Retry As Integer
    ALK_RsBaudRate As Integer
    ALK_AlBaudRate As Integer
End Type
```

VB.NET

```
Structure ALK_S_ENVIRONMENT_INFO
    Dim ALK_Ch1 As Short
    Dim ALK_Ch2 As Short
    Dim ALK_Ch3 As Short
    Dim ALK_Retry As Short
    Dim ALK_RsBaudRate As Short
    Dim ALK_AlBaudRate As Short
End Structure
```

Delphi

```
ALK_S_ENVIRONMENT_INFO = record
    ALK_Ch1: WORD;
    ALK_Ch2: WORD;
    ALK_Ch3: WORD;
    ALK_Retry: WORD;
    ALK_RsBaudRate: WORD;
    ALK_AlBaudRate: WORD;
end;
```

メンバ

ALK_Ch1 ･･･ ALK_USB が格納されます。
ALK_Ch2 ･･･ 0 が格納されます。
ALK_Ch3 ･･･ 0 が格納されます。
Retry ･･･ 環境設定ツールで設定したリトライ回数が格納されます。0 〜 3 のいずれかになります。
RsBaudRate ･･･ このメンバに格納される値は、不定となります。
AlBaudRate ･･･ 環境設定ツールで設定した AL 通信ボーレートが格納されます。
このメンバに格納される値は、次のいずれかです。

値（識別子）	値（数値）	AL 通信ボーレート
ALK_RATE_9765	0	9765bps
ALK_RATE_39062	1	39062bps
ALK_RATE_156250	2	156250bps
ALK_RATE_625000	3	625000bps

値（数値）は 10 進表記です。

スレーブ情報構造体

AL I/F ユニット対応 DLL

CB-23/USB

説　明

全スレーブのスレーブタイプを格納します。

書　式

C 言語
```
typedef struct ALK_TAG_S_SLAVE_INFO {
    WORD ALK_SlaveType[31];
} ALK_S_SLAVE_INFO;
```

VB
```
Type ALK_S_SLAVE_INFO
    ALK_SlaveType(1 To 31) As Integer
End Type
```

VB.NET
```
Structure ALK_S_SLAVE_INFO
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=31)> Public ALK_SlaveType() As Short
    Public Sub Initialize()
        ReDim ALK_SlaveType(31)
    End Sub
End Structure
```

Delphi
```
ALK_S_SLAVE_INFO = record
    ALK_SlaveType: array[1..31] of WORD;
end;
```

メンバ

次に示すメンバは、C 言語で表記しています。C 言語の *ALK_SlaveType[0]*～ *ALK_SlaveType[30]*は、Visual Basic では *ALK_SlaveType(1)*～ *ALK_SlaveType(31)*、Visual Basic.NET では *ALK_SlaveType(0)*～ *ALK_SlaveType(30)*、Delphi では *ALK_SlaveType[1]*～ *ALK_SlaveType[31]*に対応します。

ALK_SlaveType[0] ･･･ スレーブアドレス H'01 に接続されているスレーブのスレーブタイプが格納されます。

･･･

ALK_SlaveType[30] ･･･ スレーブアドレス H'1F に接続されているスレーブのスレーブタイプが格納されます。
スレーブが接続されていない場合、これらのメンバには H'FF が格納されます。
スレーブが接続されている場合、これらのメンバに格納される値は、次のいずれかです。

値（識別子）	値（数値）	スレーブタイプ
ALK_SLAVE_C770	H'00	C-770AL
ALK_SLAVE_CB08	H'10	CB-08
ALK_SLAVE_CDB770	H'20	CDB-5420-AL770
ALK_SLAVE_CAD770	H'21	CAD-5410-AL770
ALK_SLAVE_C771	H'01	C-771
ALK_SLAVE_C772	H'02	C-772
ALK_SLAVE_CD773	H'22	CD-773/ADB5331A
ALK_SLAVE_CB34	H'11	CB-34

環境設定ツール関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

環境設定ツールで設定した情報で環境設定を行います。

書　式

C 言語　BOOL ALK_EnvironmentInfo_Tool（ALK_S_RESULT FAR **psResult*）;

VB　Function ALK_EnvironmentInfo_Tool（*psResult* As ALK_S_RESULT） As Boolean

VB.NET　Function ALK_EnvironmentInfo_Tool（ByRef *psResult* As ALK_S_RESULT） As Boolean

Delphi　function ALK_EnvironmentInfo_Tool（var *psResult*: ALK_S_RESULT）: Boolean;

引　数

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

環境設定直接関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

リトライ回数、USB 通信ボーレート、AL 通信ボーレートを直接指定して環境設定を行います。

書　式

C 言語　BOOL ALK_Environment_Direct（WORD *Retry*, WORD *RsBaudRate*, WORD *AlBaudRate*, ALK_S_RESULT FAR **psResult*）;

VB　Function ALK_Environment_Direct（ByVal *Retry* As Integer, ByVal *RsBaudRate* As Integer, ByVal *AlBaudRate* As Integer, *psResult* As ALK_S_RESULT） As Boolean

VB.NET　Function ALK_Environment_Direct（ByVal *Retry* As Short, ByVal *RsBaudRate* As Short, ByVal *AlBaudRate* As Short, ByRef *psResult* As ALK_S_RESULT） As Boolean

Delphi　function ALK_Environment_Direct（*Retry*: WORD; *RsBaudRate*: WORD; *AlBaudRate*: WORD; var *psResult*: ALK_S_RESULT）: Boolean;

引　数

Retry　・・・ リトライ回数を指定します。0 ～ 3 のいずれかになります。
RsBaudRate　・・・ 必ず、NULL または 0 を指定します。
AlBaudRate　・・・ AL 通信ボーレートを指定します。

引数 *AlBaudRate* の値	AL 通信ボーレート
ALK_RATE_9765	9765bps
ALK_RATE_39062	39062bps
ALK_RATE_156250	156250bps
ALK_RATE_625000	625000bps

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

環境設定ツール情報読み出し関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

環境設定ツールで設定された情報を読み出します。

書　式

C 言語　BOOL ALK_EnvironmentInfo_Read（ALK_S_ENVIRONMENT_INFO FAR **psEnvInfo*, ALK_S_RESULT FAR **psResult*）;

VB　Function ALK_EnvironmentInfo_Read（*psEnvInfo* As ALK_S_ENVIRONMENT_INFO, *psResult* As ALK_S_RESULT） As Boolean

VB.NET　Function ALK_EnvironmentInfo_Read（ByRef *psEnvInfo* As ALK_S_ENVIRONMENT_INFO, ByRef *psResult* As ALK_S_RESULT） As Boolean

Delphi　function ALK_EnvironmentInfo_Read（var *psEnvInfo*: ALK_S_ENVIRONMENT_INFO; var *psResult*: ALK_S_RESULT）: Boolean;

引　数

psEnvInfo　･･･ 環境設定ツールで設定された情報が格納される環境設定ツール情報構造体のポインタを指定します。

psResult　･･･ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

関数環境設定問い合わせ関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

すでに環境設定が行われているかを問い合わせます。

書　式

C 言語　BOOL ALK_Environment_Inquiry（VOID）;

VB　Function ALK_Environment_Inquiry（） As Boolean

VB.NET　Function ALK_Environment_Inquiry（） As Boolean

Delphi　function ALK_Environment_Inquiry: Boolean;

引　数

この関数に、引数はありません。

戻り値

すでに環境設定が行われている場合は TRUE(1)、環境設定が行われていない場合は FALSE(0)を返します。

AL 通信エラー累計回数読み出し関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

AL 通信上で発生したエラーの累計回数を読み出します。

書　式

C 言語　BOOL ALK_Err_Count（WORD *IfNo*, WORD FAR **pCount*, ALK_S_RESULT FAR **psResult*）;

VB　Function ALK_Err_Count（ByVal *IfNo* As Integer, *pCount* As Integer, *psResult* As ALK_S_RESULT） As Boolean

VB.NET　Function ALK_Err_Count（ByVal *IfNo* As Short, ByRef *pCount* As Short, ByRef *psResult* As ALK_S_RESULT）As Boolean

Delphi　function ALK_Err_Count（*IfNo*: WORD, var *pCount*: WORD, var *psResult*: ALK_S_RESULT）: Boolean;

引　数

IfNo　・・・ I/F 番号を指定します。

マスターのボード／ユニット	引数 *IfNo* の値
CB-23/USB	ALK_USB

pCount　・・・ エラーの累計回数が格納される変数のポインタを指定します。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

AL 通信エラー累計回数クリア関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

AL 通信エラー累計回数を 0 にします。

書　式

C 言語　BOOL ALK_Err_Clear（WORD *IfNo*, ALK_S_RESULT FAR **psResult*）;

VB　Function ALK_Err_Clear（ByVal *IfNo* As Integer, *psResult* As ALK_S_RESULT） As Boolean

VB.NET　Function ALK_Err_Clear（ByVal *IfNo* As Short, ByRef *psResult* As ALK_S_RESULT） As Boolean

Delphi　function ALK_Err_Clear（*IfNo*: WORD, var *psResult*: ALK_S_RESULT）: Boolean;

引　数

IfNo　・・・ I/F 番号を指定します。

マスターのボード／ユニット	引数 *IfNo* の値
CB-23/USB	ALK_USB

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

USB 通信エラー累計回数読み出し関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

環境設定ツール関数、環境設定直接関数実行後、発生した USB 通信エラー累計回数（USB 通信エラーと USB タイムアウトエラーの累計回数）を読み出します。
カウントは最大で 65535 回までカウントし、その後エラーが発生してもカウントはストップします。
カウンタは環境設定ツール関数、環境設定直接関数、USB 通信エラー累計回数クリア関数の実行によって 0 にクリアされます。

書　式

C 言語　BOOL ALK_Usb_Err_Count（WORD FAR **pCount*, ALK_S_RESULT FAR **psResult*）;

VB　Function ALK_Usb_Err_Count（*pCount* As Integer, *psResult* As ALK_S_RESULT） As Boolean

VB.NET　Function ALK_Usb_Err_Count（ByRef *pCount* As Short, ByRef *psResult* As ALK_S_RESULT） As Boolean

Delphi　function ALK_Usb_Err_Count（var *pCount*: WORD, var *psResult*: ALK_S_RESULT）: Boolean;

引　数

pCount　・・・ エラーの累計回数が格納される変数のポインタを指定します。
psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE（1）、エラーが発生したときは FALSE（0）を返します。

USB 通信エラー累計回数クリア関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

USB 通信エラー累計回数を 0 にします。

書　式

C 言語　BOOL ALK_Usb_Err_Clear（ALK_S_RESULT FAR **psResult*）;

VB　Function ALK_Usb_Err_Clear（*psResult* As ALK_S_RESULT） As Boolean

VB.NET　Function ALK_Usb_Err_Clear（ByRef *psResult* As ALK_S_RESULT） As Boolean

Delphi　function ALK_Usb_Err_Clear（var *psResult*: ALK_S_RESULT）: Boolean;

引　数

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE（1）、エラーが発生したときは FALSE（0）を返します。

スレーブ情報読み出し関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

スレーブアドレス H'01 〜 H'1F に接続されているスレーブのスレーブタイプを読み出します。

書　式

C 言語　BOOL ALK_SlaveInfo_Read（WORD *IfNo*, ALK_S_SLAVE_INFO FAR **psSlvInfo*, ALK_S_RESULT FAR **psResult*）;

VB　Function ALK_SlaveInfo_Read（ByVal *IfNo* As Integer, *psSlvInfo* As ALK_S_SLAVE_INFO, *psResult* As ALK_S_RESULT） As Boolean

VB.NET　Function ALK_SlaveInfo_Read（ByVal *IfNo* As Short, ByRef *psSlvInfo* As ALK_S_SLAVE_INFO, ByRef *psResult* As ALK_S_RESULT） As Boolean

Delphi　function ALK_SlaveInfo_Read（*IfNo*: WORD; var *psSlvInfo*: ALK_S_SLAVE_INFO; var *psResult*: ALK_S_RESULT）: Boolean;

引　数

IfNo　・・・ I/F 番号を指定します。

マスターのボード／ユニット	引数 *IfNo* の値
CB-23/USB	ALK_USB

psSlvInfo　・・・ スレーブ情報が格納されるスレーブ情報構造体のポインタを指定します。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

5. AL I/O ユニット対応 DLL

AL I/O ユニット対応 DLL は、I/O コントロール専用のスレーブユニット上の、汎用 I/O PORT を制御するための各種関数を備えています。

この章では、AL I/O ユニット対応 DLL を、単に I/O DLL と称します。

5-1. 汎用 I/O PORT の制御

スレーブユニット上の汎用 I/O PORT を制御するために説明をします。

(1) 汎用 I/O PORT の説明

I/O DLL は、汎用 I/O PORT を直接制御します。

スレーブユニット	汎用 I/O PORT
CB-08	あり
CB-34/IO	あり（一部の信号でラッチ機能、ラッチエッジの選択が可能）

(2) 汎用 I/O PORT のオープンとクローズ

汎用 I/O PORT を制御する前に、汎用 I/O PORT をオープンし、汎用 I/O PORT ハンドルを取得します。以降、汎用 I/O PORT を制御する関数を実行する際に、この汎用 I/O PORT ハンドルを指定します。
汎用 I/O PORT ハンドルは、汎用 I/O PORT をクローズするまで有効です。ユーザアプリケーション終了時は、必ず汎用 I/O PORT をクローズしてください。クローズが行われていないと、以降正常に動作しません。

5-2. 構造体と関数

● 構造体一覧

構造体名	説明
RESULT 構造体	関数を実行した結果を格納

● 関数一覧

関数名	機能
汎用 I/O PORT オープン関数	汎用 I/O PORT のオープン
汎用 I/O PORT クローズ関数	汎用 I/O PORT のクローズ
汎用 I/O PORT 一括読み出し関数	汎用 I/O PORT の状態を一括読み出し
汎用 I/O PORT 指定信号読み出し関数	汎用 I/O PORT の指定信号の状態を読み出し
汎用 I/O PORT 一括書き込み関数	汎用 I/O PORT に一括書き込み
汎用 I/O PORT AND 書き込み関数	汎用 I/O PORT に指定データとの論理積の書き込み
汎用 I/O PORT OR 書き込み関数	汎用 I/O PORT に指定データとの論理和を書き込み
汎用 I/O PORT 指定信号 LOW レベル書き込み関数	汎用 I/O PORT の指定信号に LOW レベルを書き込み
汎用 I/O PORT 指定信号 HIGH レベル書き込み関数	汎用 I/O PORT の指定信号に HIGH レベルを書き込み
ラッチ機能選択書き込み関数	汎用 I/O PORT のラッチ機能を設定
ラッチエッジ選択書き込み関数	汎用 I/O PORT のラッチエッジを設定
ラッチ機能選択読み出し関数	汎用 I/O PORT のラッチ機能の設定の読み出し
ラッチエッジ選択読み出し関数	汎用 I/O PORT のラッチエッジの設定の読み出し
ラッチクリア書き込み関数	汎用 I/O PORT のラッチデータをクリア

RESULT 構造体

AL I/O ユニット対応 DLL

CB-08　CB-34/IO

説　明

I/O DLL の関数を実行した結果が格納されます。

書　式

C 言語
```
typedef struct ACIO_TAG_S_RESULT {
    WORD   ACIO_Result[4];
} ACIO_S_RESULT;
```

VB
```
Type ACIO_S_RESULT
    ACIO_Result(1 To 4) As Interger
End Type
```

VB.NET
```
Structure ACIO_S_RESULT
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=4)> Public ACIO_Result() As Short
    Public Sub Initialize()
        ReDim ACIO_Result(4)
    End Sub
End Structure
```

Delphi
```
ACIO_S_RESULT = record
    ACIO_Result: array[1..4] of WORD;
end;
```

メンバ

次に示すメンバは、C 言語で表記しています。C 言語の *ACIO_Result[0]*～ *ACIO_Result[3]*は、Visual Basic では *ACIO_Result(1)*～ *ACIO_Result(4)*、Visual Basic.NET では *ACIO_Result(0)*～ *ACIO_Result(3)*、Delphi では *ACIO_Result[1]*～ *ACIO_Result[4]*に対応します。

ACIO_Result[0]　・・・ 実行された関数を示します。このメンバに格納される値は、次のいずれかです。

値	実行された関数	値	実行された関数
60	汎用 I/O PORT オープン関数	130	ラッチ機能選択書き込み関数
61	汎用 I/O PORT クローズ関数	131	ラッチエッジ選択書き込み関数
62	汎用 I/O PORT 一括読み出し関数	132	ラッチ機能選択読み出し関数
63	汎用 I/O PORT 指定信号読み出し関数	133	ラッチエッジ選択読み出し関数
64	汎用 I/O PORT 一括書き込み関数	134	ラッチクリア書き込み関数
65	汎用 I/O PORT AND 書き込み関数		
66	汎用 I/O PORT OR 書き込み関数		
67	汎用 I/O PORT 指定信号 LOW 書き込み関数		
68	汎用 I/O PORT 指定信号 HIGH 書き込み関数		

値は 10 進表記です。

ACIO_Result[1]　・・・ 実行結果を示します。このメンバに格納される値は、次のいずれかです。

値	実行結果
0	関数の実行が正常に終了しました
1	DLL 内部で API エラーが発生しました
2	NULL ポインタが指定されています
4	指定したパラメータが設定範囲を超えています
6	ハンドルの値が範囲外です
8	環境設定されていません
9	指定されたアドレスにスレーブが接続されていません
10	指定したスレーブタイプと実際に接続されているスレーブのタイプが異なります
11	接続されているスレーブでは、関数の実行ができません
12	指定された汎用 I/O PORT は、オープンされていません
13	指定された汎用 I/O PORT は、すでにオープンされています
18	I/O PORT 数が最大値を越えています
29	現在のスレッドからは、操作出来ません
30	I/F ユニット DLL が見つかりません

値は 10 進表記です。

ACIO_Result[2]　・・・ I/F ユニット、スレーブユニット上で発生したエラーを示します。
このメンバに格納される値は、次のいずれかです。

値		実行結果
0	（H'00）	エラーは発生しませんでした
1 ～ 127	（H'01 ～ H'7F）	スレーブユニット固有のエラーです。各スレーブの取扱説明書をご覧ください
128	（H'80h）	初期化エラー（スレーブユニットが不正に電源 OFF または RESET されました）
129	（H'81h）	シリアルエラー（スレーブユニットからの受信時にエラーが発生しました）
130	（H'82h）	タイムアウトエラー（スレーブユニットへの送信時にエラーが発生しました）

値は 10 進表記です。()内は、16 進表記です。

ACIO_Result[3]　・・・ 将来の拡張用です。

汎用 I/O PORT オープン関数

AL I/O ユニット対応 DLL

CB-08 CB-34/IO

機　能

I/F 番号、スレーブアドレス、スレーブタイプを指定して、汎用 I/O PORT をオープンし、引数 *phPort* で示される変数に汎用 I/O PORT ハンドルを格納します。

書　式

C 言語　BOOL ACIO_BPortOpen（WORD *IfNo*, WORD *SlaveAddr*, WORD *SlaveType*, WORD *Sel*, DWORD FAR **phPort*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BPortOpen（ByVal *IfNo* As Integer, ByVal *SlaveAddr* As Integer, ByVal *SlaveType* As Integer, ByVal *Sel* As Integer, *phPort* As Long, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BPortOpen（ByVal *IfNo* As Short, ByVal *SlaveAddr* As Short, ByVal *SlaveType* As Short, ByVal *Sel* As Short, ByRef *phPort* As Integer, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BPortOpen（*IfNo*: WORD; *SlaveAddr*: WORD; *SlaveType*: WORD; *Sel*: WORD; var *phPort*: DWORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

IfNo　・・・ I/F 番号を指定します。

マスターのボード／ユニット	引数 *IfNo* の値
CB-09	ボード番号（0 ～ 9）
CB-23/USB	ACIO_USB

SlaveAddr　・・・ スレーブアドレスを指定します。H'01 ～ H'1F のいずれかになります。
SlaveType　・・・ スレーブタイプを指定します。この引数は、次の値のいずれかになります。

引数 *SlaveType* の値	スレーブタイプ
ACIO_SLAVE_CB08	CB-08
ACIO_SLAVE_CB34	CB-34/IO

Sel　・・・ オープンする PORT を指定します。CB-34/IO のみで有効です。

引数 *Select* の値	オープンする PORT
ACIO_CB34_10_20	$\overline{\text{IN10}}$ ～ $\overline{\text{IN17}}$、$\overline{\text{IN20}}$ ～ $\overline{\text{IN27}}$、$\overline{\text{OUT10}}$ ～ $\overline{\text{OUT17}}$、$\overline{\text{OUT20}}$ ～ $\overline{\text{OUT27}}$
ACIO_CB34_30_40	$\overline{\text{IN30}}$ ～ $\overline{\text{IN37}}$、$\overline{\text{IN40}}$ ～ $\overline{\text{IN47}}$、$\overline{\text{OUT30}}$ ～ $\overline{\text{OUT37}}$、$\overline{\text{OUT40}}$ ～ $\overline{\text{OUT47}}$

phPort　・・・ 汎用 I/O PORT ハンドルが格納される変数のポインタを指定します。
psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT クローズ関数

AL I/O ユニット対応 DLL

CB-08 CB-34/IO

機　能

指定された汎用 I/O PORT をクローズします。

書　式

C 言語　BOOL ACIO_BPortClose（DWORD *hPort*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BPortClose（ByVal *hPort* As Long, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BPortClose（ByVal *hPort* As Integer, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BPortClose（*hPort*: DWORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

hPort　･･･ 汎用 I/O PORT ハンドルを指定します。
psResult　･･･ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT **一括読み出し関数**

AL I/O **ユニット対応** DLL

CB-08 CB-34/IO

機　能

指定された汎用 I/O PORT の状態を一括読み出しします。

書　式

C 言語　BOOL ACIO_BPortIn（DWORD *hPort*, DWORD FAR **pData*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BPortIn（ByVal *hPort* As Long, *pData* As Long, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BPortIn（ByVal *hPort* As Integer, ByRef *pData* As Integer, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BPortIn（*hPort*: DWORD; var *pData*: DWORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

hPort　・・・ 汎用 I/O PORT ハンドルを指定します。

pData　・・・ 読み出した内容が格納される変数のポインタを指定します。
変数の内容は、次のようになります。

● CB-08 **の汎用** I/O PORT **を読み出す場合**

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
$\overline{\text{IN27}}$	$\overline{\text{IN26}}$	$\overline{\text{IN25}}$	$\overline{\text{IN24}}$	$\overline{\text{IN23}}$	$\overline{\text{IN22}}$	$\overline{\text{IN21}}$	$\overline{\text{IN20}}$	$\overline{\text{IN17}}$	$\overline{\text{IN16}}$	$\overline{\text{IN15}}$	$\overline{\text{IN14}}$	$\overline{\text{IN13}}$	$\overline{\text{IN12}}$	$\overline{\text{IN11}}$	$\overline{\text{IN10}}$

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{\text{OUT27}}$	$\overline{\text{OUT26}}$	$\overline{\text{OUT25}}$	$\overline{\text{OUT24}}$	$\overline{\text{OUT23}}$	$\overline{\text{OUT22}}$	$\overline{\text{OUT21}}$	$\overline{\text{OUT20}}$	$\overline{\text{OUT17}}$	$\overline{\text{OUT16}}$	$\overline{\text{OUT15}}$	$\overline{\text{OUT14}}$	$\overline{\text{OUT13}}$	$\overline{\text{OUT12}}$	$\overline{\text{OUT11}}$	$\overline{\text{OUT10}}$

HIGH **レベルのときに** 0、LOW **レベルのときに** 1 **が読み出されます。**

● CB-34/IO **の汎用** I/O PORT **を読み出す場合**

・$\overline{\text{IN10}}$ **〜** $\overline{\text{IN17}}$、$\overline{\text{IN20}}$ **〜** $\overline{\text{IN27}}$、$\overline{\text{OUT10}}$ **〜** $\overline{\text{OUT17}}$、$\overline{\text{OUT20}}$ **〜** $\overline{\text{OUT27}}$ **をオープンしている場合**

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
$\overline{\text{IN27}}$	$\overline{\text{IN26}}$	$\overline{\text{IN25}}$	$\overline{\text{IN24}}$	$\overline{\text{IN23}}$	$\overline{\text{IN22}}$	$\overline{\text{IN21}}$	$\overline{\text{IN20}}$	$\overline{\text{IN17}}$	$\overline{\text{IN16}}$	$\overline{\text{IN15}}$	$\overline{\text{IN14}}$	$\overline{\text{IN13}}$	$\overline{\text{IN12}}$	$\overline{\text{IN11}}$	$\overline{\text{IN10}}$

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{\text{OUT27}}$	$\overline{\text{OUT26}}$	$\overline{\text{OUT25}}$	$\overline{\text{OUT24}}$	$\overline{\text{OUT23}}$	$\overline{\text{OUT22}}$	$\overline{\text{OUT21}}$	$\overline{\text{OUT20}}$	$\overline{\text{OUT17}}$	$\overline{\text{OUT16}}$	$\overline{\text{OUT15}}$	$\overline{\text{OUT14}}$	$\overline{\text{OUT13}}$	$\overline{\text{OUT12}}$	$\overline{\text{OUT11}}$	$\overline{\text{OUT10}}$

HIGH **レベルのときに** 0、LOW **レベルのときに** 1 **が読み出されます。**

・$\overline{\text{IN30}}$ **〜** $\overline{\text{IN37}}$、$\overline{\text{IN40}}$ **〜** $\overline{\text{IN47}}$、$\overline{\text{OUT30}}$ **〜** $\overline{\text{OUT37}}$、$\overline{\text{OUT40}}$ **〜** $\overline{\text{OUT47}}$ **をオープンしている場合**

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
$\overline{\text{IN47}}$	$\overline{\text{IN46}}$	$\overline{\text{IN45}}$	$\overline{\text{IN44}}$	$\overline{\text{IN43}}$	$\overline{\text{IN42}}$	$\overline{\text{IN41}}$	$\overline{\text{IN40}}$	$\overline{\text{IN37}}$	$\overline{\text{IN36}}$	$\overline{\text{IN35}}$	$\overline{\text{IN34}}$	$\overline{\text{IN33}}$	$\overline{\text{IN32}}$	$\overline{\text{IN31}}$	$\overline{\text{IN30}}$

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{\text{OUT47}}$	$\overline{\text{OUT46}}$	$\overline{\text{OUT45}}$	$\overline{\text{OUT44}}$	$\overline{\text{OUT43}}$	$\overline{\text{OUT42}}$	$\overline{\text{OUT41}}$	$\overline{\text{OUT40}}$	$\overline{\text{OUT37}}$	$\overline{\text{OUT36}}$	$\overline{\text{OUT35}}$	$\overline{\text{OUT34}}$	$\overline{\text{OUT33}}$	$\overline{\text{OUT32}}$	$\overline{\text{OUT31}}$	$\overline{\text{OUT30}}$

HIGH **レベルのときに** 0、LOW **レベルのときに** 1 **が読み出されます。**

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT 指定信号読み出し関数

AL I/O ユニット対応 DLL

CB-08　CB-34/IO

機　能

指定された汎用 I/O PORT の指定信号の状態を読み出します。

書　式

C 言語　BOOL ACIO_BSigIn（DWORD *hPort*, WORD *SigNo*, WORD FAR **pStatus*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BSigIn（ByVal *hPort* As Long, ByVal *SigNo* As Integer, *pStatus* As Integer, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BSigIn（ByVal *hPort* As Integer, ByVal *SigNo* As Short, ByRef *pStatus* As Short, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BSigIn（*hPort*: DWORD; *SigNo*: WORD; var *pStatus*: WORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

hPort　・・・ 汎用 I/O PORT ハンドルを指定します。

SigNo　・・・ 信号を指定します。

● CB-08 の汎用 I/O PORT を読み出す場合

引数 *SigNo* の値	指定信号	引数 *SigNo* の値	指定信号
ACIO_CB08_IN10	$\overline{IN10}$	ACIO_CB08_IN20	$\overline{IN20}$
ACIO_CB08_IN11	$\overline{IN11}$	ACIO_CB08_IN21	$\overline{IN21}$
ACIO_CB08_IN12	$\overline{IN12}$	ACIO_CB08_IN22	$\overline{IN22}$
ACIO_CB08_IN13	$\overline{IN13}$	ACIO_CB08_IN23	$\overline{IN23}$
ACIO_CB08_IN14	$\overline{IN14}$	ACIO_CB08_IN24	$\overline{IN24}$
ACIO_CB08_IN15	$\overline{IN15}$	ACIO_CB08_IN25	$\overline{IN25}$
ACIO_CB08_IN16	$\overline{IN16}$	ACIO_CB08_IN26	$\overline{IN26}$
ACIO_CB08_IN17	$\overline{IN17}$	ACIO_CB08_IN27	$\overline{IN27}$

● CB-34/IO の汎用 I/O PORT を読み出す場合

・$\overline{IN10}$ ～ $\overline{IN17}$、$\overline{IN20}$ ～ $\overline{IN27}$、$\overline{OUT10}$ ～ $\overline{OUT17}$、$\overline{OUT20}$ ～ $\overline{OUT27}$ をオープンしている場合

引数 *SigNo* の値	指定信号	引数 *SigNo* の値	指定信号
ACIO_CB34_IN10	$\overline{IN10}$	ACIO_CB34_IN20	$\overline{IN20}$
ACIO_CB34_IN11	$\overline{IN11}$	ACIO_CB34_IN21	$\overline{IN21}$
ACIO_CB34_IN12	$\overline{IN12}$	ACIO_CB34_IN22	$\overline{IN22}$
ACIO_CB34_IN13	$\overline{IN13}$	ACIO_CB34_IN23	$\overline{IN23}$
ACIO_CB34_IN14	$\overline{IN14}$	ACIO_CB34_IN24	$\overline{IN24}$
ACIO_CB34_IN15	$\overline{IN15}$	ACIO_CB34_IN25	$\overline{IN25}$
ACIO_CB34_IN16	$\overline{IN16}$	ACIO_CB34_IN26	$\overline{IN26}$
ACIO_CB34_IN17	$\overline{IN17}$	ACIO_CB34_IN27	$\overline{IN27}$

・$\overline{IN30}$ ～ $\overline{IN37}$、$\overline{IN40}$ ～ $\overline{IN47}$、$\overline{OUT30}$ ～ $\overline{OUT37}$、$\overline{OUT40}$ ～ $\overline{OUT47}$ をオープンしている場合

引数 *SigNo* の値	指定信号	引数 *SigNo* の値	指定信号
ACIO_CB34_IN30	$\overline{IN30}$	ACIO_CB34_IN40	$\overline{IN40}$
ACIO_CB34_IN31	$\overline{IN31}$	ACIO_CB34_IN41	$\overline{IN41}$
ACIO_CB34_IN32	$\overline{IN32}$	ACIO_CB34_IN42	$\overline{IN42}$
ACIO_CB34_IN33	$\overline{IN33}$	ACIO_CB34_IN43	$\overline{IN43}$
ACIO_CB34_IN34	$\overline{IN34}$	ACIO_CB34_IN44	$\overline{IN44}$
ACIO_CB34_IN35	$\overline{IN35}$	ACIO_CB34_IN45	$\overline{IN45}$
ACIO_CB34_IN36	$\overline{IN36}$	ACIO_CB34_IN46	$\overline{IN46}$
ACIO_CB34_IN37	$\overline{IN37}$	ACIO_CB34_IN47	$\overline{IN47}$

pStatus　・・・ 指定信号の状態が格納される変数のポインタを指定します。

格納される値	指定信号のレベル
ACIO_LOW	LOW レベル
ACIO_HIGH	HIGH レベル

psResult　　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT 一括書き込み関数

AL I/O ユニット対応 DLL

CB-08 CB-34/IO

機　能

指定された汎用 I/O PORT にデータを一括書き込みします。

書　式

C 言語　BOOL ACIO_BPortOut（DWORD *hPort*, DWORD FAR **pData*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BPortOut（ByVal *hPort* As Long, *pData* As Long, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BPortOut（ByVal *hPort* As Integer, ByRef *pData* As Integer, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BPortOut（*hPort*: DWORD; var *pData*: DWORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

hPort　・・・ 汎用 I/O PORT ハンドルを指定します。

pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
変数の内容は、次のようになります。

● CB-08 の汎用 I/O PORT に書き込む場合

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{\text{OUT27}}$	$\overline{\text{OUT26}}$	$\overline{\text{OUT25}}$	$\overline{\text{OUT24}}$	$\overline{\text{OUT23}}$	$\overline{\text{OUT22}}$	$\overline{\text{OUT21}}$	$\overline{\text{OUT20}}$	$\overline{\text{OUT17}}$	$\overline{\text{OUT16}}$	$\overline{\text{OUT15}}$	$\overline{\text{OUT14}}$	$\overline{\text{OUT13}}$	$\overline{\text{OUT12}}$	$\overline{\text{OUT11}}$	$\overline{\text{OUT10}}$

HIGH レベルにするときに 0、LOW レベルにするときに 1 を書き込みます。

● CB-34/IO の汎用 I/O PORT に書き込む場合

・$\overline{\text{IN10}}$ 〜 $\overline{\text{IN17}}$、$\overline{\text{IN20}}$ 〜 $\overline{\text{IN27}}$、$\overline{\text{OUT10}}$ 〜 $\overline{\text{OUT17}}$、$\overline{\text{OUT20}}$ 〜 $\overline{\text{OUT27}}$ をオープンしている場合

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{\text{OUT27}}$	$\overline{\text{OUT26}}$	$\overline{\text{OUT25}}$	$\overline{\text{OUT24}}$	$\overline{\text{OUT23}}$	$\overline{\text{OUT22}}$	$\overline{\text{OUT21}}$	$\overline{\text{OUT20}}$	$\overline{\text{OUT17}}$	$\overline{\text{OUT16}}$	$\overline{\text{OUT15}}$	$\overline{\text{OUT14}}$	$\overline{\text{OUT13}}$	$\overline{\text{OUT12}}$	$\overline{\text{OUT11}}$	$\overline{\text{OUT10}}$

HIGH レベルにするときに 0、LOW レベルにするときに 1 を書き込みます。

・$\overline{\text{IN30}}$ 〜 $\overline{\text{IN37}}$、$\overline{\text{IN40}}$ 〜 $\overline{\text{IN47}}$、$\overline{\text{OUT30}}$ 〜 $\overline{\text{OUT37}}$、$\overline{\text{OUT40}}$ 〜 $\overline{\text{OUT47}}$ をオープンしている場合

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{\text{OUT47}}$	$\overline{\text{OUT46}}$	$\overline{\text{OUT45}}$	$\overline{\text{OUT44}}$	$\overline{\text{OUT43}}$	$\overline{\text{OUT42}}$	$\overline{\text{OUT41}}$	$\overline{\text{OUT40}}$	$\overline{\text{OUT37}}$	$\overline{\text{OUT36}}$	$\overline{\text{OUT35}}$	$\overline{\text{OUT34}}$	$\overline{\text{OUT33}}$	$\overline{\text{OUT32}}$	$\overline{\text{OUT31}}$	$\overline{\text{OUT30}}$

HIGH レベルにするときに 0、LOW レベルにするときに 1 を書き込みます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT AND 書き込み関数

AL I/O ユニット対応 DLL

CB-08 CB-34/IO

機　能

指定された汎用 I/O PORT の内容と、指定された変数の内容の論理積を、指定された汎用 I/O PORT に書き込みます。

書　式

C 言語　BOOL ACIO_BPortAndOut（DWORD *hPort*, DWORD FAR **pData*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BPortAndOut（ByVal *hPort* As Long, *pData* As Long, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BPortAndOut（ByVal *hPort* As Integer, ByRef *pData* As Integer, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BPortAndOut（*hPort*: DWORD; var *pData*: DWORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

hPort　・・・ 汎用 I/O PORT ハンドルを指定します。

pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
変数の内容は、次のようになります。

● CB-08 の汎用 I/O PORT に書き込む場合

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{OUT27}$	$\overline{OUT26}$	$\overline{OUT25}$	$\overline{OUT24}$	$\overline{OUT23}$	$\overline{OUT22}$	$\overline{OUT21}$	$\overline{OUT20}$	$\overline{OUT17}$	$\overline{OUT16}$	$\overline{OUT15}$	$\overline{OUT14}$	$\overline{OUT13}$	$\overline{OUT12}$	$\overline{OUT11}$	$\overline{OUT10}$

HIGH レベルにするときに 0、変更しないときに 1 を書き込みます。

● CB-34/IO の汎用 I/O PORT に書き込む場合

・$\overline{IN10}$ 〜 $\overline{IN17}$、$\overline{IN20}$ 〜 $\overline{IN27}$、$\overline{OUT10}$ 〜 $\overline{OUT17}$、$\overline{OUT20}$ 〜 $\overline{OUT27}$ をオープンしている場合

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{OUT27}$	$\overline{OUT26}$	$\overline{OUT25}$	$\overline{OUT24}$	$\overline{OUT23}$	$\overline{OUT22}$	$\overline{OUT21}$	$\overline{OUT20}$	$\overline{OUT17}$	$\overline{OUT16}$	$\overline{OUT15}$	$\overline{OUT14}$	$\overline{OUT13}$	$\overline{OUT12}$	$\overline{OUT11}$	$\overline{OUT10}$

HIGH レベルにするときに 0、変更しないときに 1 を書き込みます。

・$\overline{IN30}$ 〜 $\overline{IN37}$、$\overline{IN40}$ 〜 $\overline{IN47}$、$\overline{OUT30}$ 〜 $\overline{OUT37}$、$\overline{OUT40}$ 〜 $\overline{OUT47}$ をオープンしている場合

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{OUT47}$	$\overline{OUT46}$	$\overline{OUT45}$	$\overline{OUT44}$	$\overline{OUT43}$	$\overline{OUT42}$	$\overline{OUT41}$	$\overline{OUT40}$	$\overline{OUT37}$	$\overline{OUT36}$	$\overline{OUT35}$	$\overline{OUT34}$	$\overline{OUT33}$	$\overline{OUT32}$	$\overline{OUT31}$	$\overline{OUT30}$

HIGH レベルにするときに 0、変更しないときに 1 を書き込みます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT OR 書き込み関数

AL I/O ユニット対応 DLL

CB-08　CB-34/IO

機　能

指定された汎用 I/O PORT の内容と、指定された変数の内容の論理和を、指定された汎用 I/O PORT に書き込みます。

書　式

C 言語　BOOL ACIO_BPortOrOut（DWORD *hPort*, DWORD FAR **pData*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BPortOrOut（ByVal *hPort* As Long, *pData* As Long, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BPortOrOut（ByVal *hPort* As Integer, ByRef *pData* As Integer, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BPortOrOut（*hPort*: DWORD; var *pData*: DWORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

hPort　･･･ 汎用 I/O PORT ハンドルを指定します。

pData　･･･ 書き込むデータが格納されている変数のポインタを指定します。
変数の内容は、次のようになります。

● CB-08 の汎用 I/O PORT に書き込む場合

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{\text{OUT27}}$	$\overline{\text{OUT26}}$	$\overline{\text{OUT25}}$	$\overline{\text{OUT24}}$	$\overline{\text{OUT23}}$	$\overline{\text{OUT22}}$	$\overline{\text{OUT21}}$	$\overline{\text{OUT20}}$	$\overline{\text{OUT17}}$	$\overline{\text{OUT16}}$	$\overline{\text{OUT15}}$	$\overline{\text{OUT14}}$	$\overline{\text{OUT13}}$	$\overline{\text{OUT12}}$	$\overline{\text{OUT11}}$	$\overline{\text{OUT10}}$

変更しないときに 0、LOW レベルにするときに 1 を書き込みます。

● CB-34/IO の汎用 I/O PORT に書き込む場合

・$\overline{\text{IN10}}$ ～ $\overline{\text{IN17}}$、$\overline{\text{IN20}}$ ～ $\overline{\text{IN27}}$、$\overline{\text{OUT10}}$ ～ $\overline{\text{OUT17}}$、$\overline{\text{OUT20}}$ ～ $\overline{\text{OUT27}}$ をオープンしている場合

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{\text{OUT27}}$	$\overline{\text{OUT26}}$	$\overline{\text{OUT25}}$	$\overline{\text{OUT24}}$	$\overline{\text{OUT23}}$	$\overline{\text{OUT22}}$	$\overline{\text{OUT21}}$	$\overline{\text{OUT20}}$	$\overline{\text{OUT17}}$	$\overline{\text{OUT16}}$	$\overline{\text{OUT15}}$	$\overline{\text{OUT14}}$	$\overline{\text{OUT13}}$	$\overline{\text{OUT12}}$	$\overline{\text{OUT11}}$	$\overline{\text{OUT10}}$

変更しないときに 0、LOW レベルにするときに 1 を書き込みます。

・$\overline{\text{IN30}}$ ～ $\overline{\text{IN37}}$、$\overline{\text{IN40}}$ ～ $\overline{\text{IN47}}$、$\overline{\text{OUT30}}$ ～ $\overline{\text{OUT37}}$、$\overline{\text{OUT40}}$ ～ $\overline{\text{OUT47}}$ をオープンしている場合

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$\overline{\text{OUT47}}$	$\overline{\text{OUT46}}$	$\overline{\text{OUT45}}$	$\overline{\text{OUT44}}$	$\overline{\text{OUT43}}$	$\overline{\text{OUT42}}$	$\overline{\text{OUT41}}$	$\overline{\text{OUT40}}$	$\overline{\text{OUT37}}$	$\overline{\text{OUT36}}$	$\overline{\text{OUT35}}$	$\overline{\text{OUT34}}$	$\overline{\text{OUT33}}$	$\overline{\text{OUT32}}$	$\overline{\text{OUT31}}$	$\overline{\text{OUT30}}$

変更しないときに 0、LOW レベルにするときに 1 を書き込みます。

psResult　･･･ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT 指定信号 LOW レベル書き込み関数

AL I/O ユニット対応 DLL

CB-08 CB-34/IO

機　能
指定された汎用 I/O PORT の指定信号を LOW レベルにします。

書　式
C 言語　BOOL ACIO_BSigLowOut（DWORD *hPort*, WORD *SigNo*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BSigLowOut（ByVal *hPort* As Long, ByVal *SigNo* As Integer, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BSigLowOut（ByVal *hPort* As Integer, ByVal *SigNo* As Short, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BSigLowOut（*hPort*: DWORD; *SigNo*: WORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数
hPort　・・・ 汎用 I/O PORT ハンドルを指定します。

SigNo　・・・ 信号を指定します。

● CB-08 の汎用 I/O PORT に書き込む場合

引数 *SigNo* の値	指定信号	引数 *SigNo* の値	指定信号
ACIO_CB08_OUT10	$\overline{\mathrm{OUT10}}$	ACIO_CB08_OUT20	$\overline{\mathrm{OUT20}}$
ACIO_CB08_OUT11	$\overline{\mathrm{OUT11}}$	ACIO_CB08_OUT21	$\overline{\mathrm{OUT21}}$
ACIO_CB08_OUT12	$\overline{\mathrm{OUT12}}$	ACIO_CB08_OUT22	$\overline{\mathrm{OUT22}}$
ACIO_CB08_OUT13	$\overline{\mathrm{OUT13}}$	ACIO_CB08_OUT23	$\overline{\mathrm{OUT23}}$
ACIO_CB08_OUT14	$\overline{\mathrm{OUT14}}$	ACIO_CB08_OUT24	$\overline{\mathrm{OUT24}}$
ACIO_CB08_OUT15	$\overline{\mathrm{OUT15}}$	ACIO_CB08_OUT25	$\overline{\mathrm{OUT25}}$
ACIO_CB08_OUT16	$\overline{\mathrm{OUT16}}$	ACIO_CB08_OUT26	$\overline{\mathrm{OUT26}}$
ACIO_CB08_OUT17	$\overline{\mathrm{OUT17}}$	ACIO_CB08_OUT27	$\overline{\mathrm{OUT27}}$

● CB-34/IO の汎用 I/O PORT に書き込む場合

・$\overline{\mathrm{IN10}}$ ～ $\overline{\mathrm{IN17}}$、$\overline{\mathrm{IN20}}$ ～ $\overline{\mathrm{IN27}}$、$\overline{\mathrm{OUT10}}$ ～ $\overline{\mathrm{OUT17}}$、$\overline{\mathrm{OUT20}}$ ～ $\overline{\mathrm{OUT27}}$ をオープンしている場合

引数 *SigNo* の値	指定信号	引数 *SigNo* の値	指定信号
ACIO_CB34_OUT10	$\overline{\mathrm{OUT10}}$	ACIO_CB34_OUT20	$\overline{\mathrm{OUT20}}$
ACIO_CB34_OUT11	$\overline{\mathrm{OUT11}}$	ACIO_CB34_OUT21	$\overline{\mathrm{OUT21}}$
ACIO_CB34_OUT12	$\overline{\mathrm{OUT12}}$	ACIO_CB34_OUT22	$\overline{\mathrm{OUT22}}$
ACIO_CB34_OUT13	$\overline{\mathrm{OUT13}}$	ACIO_CB34_OUT23	$\overline{\mathrm{OUT23}}$
ACIO_CB34_OUT14	$\overline{\mathrm{OUT14}}$	ACIO_CB34_OUT24	$\overline{\mathrm{OUT24}}$
ACIO_CB34_OUT15	$\overline{\mathrm{OUT15}}$	ACIO_CB34_OUT25	$\overline{\mathrm{OUT25}}$
ACIO_CB34_OUT16	$\overline{\mathrm{OUT16}}$	ACIO_CB34_OUT26	$\overline{\mathrm{OUT26}}$
ACIO_CB34_OUT17	$\overline{\mathrm{OUT17}}$	ACIO_CB34_OUT27	$\overline{\mathrm{OUT27}}$

・$\overline{\mathrm{IN30}}$ ～ $\overline{\mathrm{IN37}}$、$\overline{\mathrm{IN40}}$ ～ $\overline{\mathrm{IN47}}$、$\overline{\mathrm{OUT30}}$ ～ $\overline{\mathrm{OUT37}}$、$\overline{\mathrm{OUT40}}$ ～ $\overline{\mathrm{OUT47}}$ をオープンしている場合

引数 *SigNo* の値	指定信号	引数 *SigNo* の値	指定信号
ACIO_CB34_OUT30	$\overline{\mathrm{OUT30}}$	ACIO_CB34_OUT40	$\overline{\mathrm{OUT40}}$
ACIO_CB34_OUT31	$\overline{\mathrm{OUT31}}$	ACIO_CB34_OUT41	$\overline{\mathrm{OUT41}}$
ACIO_CB34_OUT32	$\overline{\mathrm{OUT32}}$	ACIO_CB34_OUT42	$\overline{\mathrm{OUT42}}$
ACIO_CB34_OUT33	$\overline{\mathrm{OUT33}}$	ACIO_CB34_OUT43	$\overline{\mathrm{OUT43}}$
ACIO_CB34_OUT34	$\overline{\mathrm{OUT34}}$	ACIO_CB34_OUT44	$\overline{\mathrm{OUT44}}$
ACIO_CB34_OUT35	$\overline{\mathrm{OUT35}}$	ACIO_CB34_OUT45	$\overline{\mathrm{OUT45}}$
ACIO_CB34_OUT36	$\overline{\mathrm{OUT36}}$	ACIO_CB34_OUT46	$\overline{\mathrm{OUT46}}$
ACIO_CB34_OUT37	$\overline{\mathrm{OUT37}}$	ACIO_CB34_OUT47	$\overline{\mathrm{OUT47}}$

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値
この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT 指定信号 HIGH レベル書き込み関数

AL I/O ユニット対応 DLL

CB-08　CB-34/IO

機　能

指定された汎用 I/O PORT の指定信号を HIGH レベルにします。

書　式

C 言語　BOOL ACIO_BSigHighOut（DWORD *hPort*, WORD *SigNo*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BSigHighOut（ByVal *hPort* As Long, ByVal *SigNo* As Integer, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BSigHighOut（ByVal *hPort* As Integer, ByVal *SigNo* As Short, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BSigHighOut（*hPort*: DWORD; *SigNo*: WORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

hPort　・・・ 汎用 I/O PORT ハンドルを指定します。

SigNo　・・・ 信号を指定します。

● CB-08 の汎用 I/O PORT に書き込む場合

引数 *SigNo* の値	指定信号	引数 *SigNo* の値	指定信号
ACIO_CB08_OUT10	$\overline{OUT10}$	ACIO_CB08_OUT20	$\overline{OUT20}$
ACIO_CB08_OUT11	$\overline{OUT11}$	ACIO_CB08_OUT21	$\overline{OUT21}$
ACIO_CB08_OUT12	$\overline{OUT12}$	ACIO_CB08_OUT22	$\overline{OUT22}$
ACIO_CB08_OUT13	$\overline{OUT13}$	ACIO_CB08_OUT23	$\overline{OUT23}$
ACIO_CB08_OUT14	$\overline{OUT14}$	ACIO_CB08_OUT24	$\overline{OUT24}$
ACIO_CB08_OUT15	$\overline{OUT15}$	ACIO_CB08_OUT25	$\overline{OUT25}$
ACIO_CB08_OUT16	$\overline{OUT16}$	ACIO_CB08_OUT26	$\overline{OUT26}$
ACIO_CB08_OUT17	$\overline{OUT17}$	ACIO_CB08_OUT27	$\overline{OUT27}$

● CB-34/IO の汎用 I/O PORT に書き込む場合

・$\overline{IN10}$ ～ $\overline{IN17}$、$\overline{IN20}$ ～ $\overline{IN27}$、$\overline{OUT10}$ ～ $\overline{OUT17}$、$\overline{OUT20}$ ～ $\overline{OUT27}$ をオープンしている場合

引数 *SigNo* の値	指定信号	引数 *SigNo* の値	指定信号
ACIO_CB34_OUT10	$\overline{OUT10}$	ACIO_CB34_OUT20	$\overline{OUT20}$
ACIO_CB34_OUT11	$\overline{OUT11}$	ACIO_CB34_OUT21	$\overline{OUT21}$
ACIO_CB34_OUT12	$\overline{OUT12}$	ACIO_CB34_OUT22	$\overline{OUT22}$
ACIO_CB34_OUT13	$\overline{OUT13}$	ACIO_CB34_OUT23	$\overline{OUT23}$
ACIO_CB34_OUT14	$\overline{OUT14}$	ACIO_CB34_OUT24	$\overline{OUT24}$
ACIO_CB34_OUT15	$\overline{OUT15}$	ACIO_CB34_OUT25	$\overline{OUT25}$
ACIO_CB34_OUT16	$\overline{OUT16}$	ACIO_CB34_OUT26	$\overline{OUT26}$
ACIO_CB34_OUT17	$\overline{OUT17}$	ACIO_CB34_OUT27	$\overline{OUT27}$

・$\overline{IN30}$ ～ $\overline{IN37}$、$\overline{IN40}$ ～ $\overline{IN47}$、$\overline{OUT30}$ ～ $\overline{OUT37}$、$\overline{OUT40}$ ～ $\overline{OUT47}$ をオープンしている場合

引数 *SigNo* の値	指定信号	引数 *SigNo* の値	指定信号
ACIO_CB34_OUT30	$\overline{OUT30}$	ACIO_CB34_OUT40	$\overline{OUT40}$
ACIO_CB34_OUT31	$\overline{OUT31}$	ACIO_CB34_OUT41	$\overline{OUT41}$
ACIO_CB34_OUT32	$\overline{OUT32}$	ACIO_CB34_OUT42	$\overline{OUT42}$
ACIO_CB34_OUT33	$\overline{OUT33}$	ACIO_CB34_OUT43	$\overline{OUT43}$
ACIO_CB34_OUT34	$\overline{OUT34}$	ACIO_CB34_OUT44	$\overline{OUT44}$
ACIO_CB34_OUT35	$\overline{OUT35}$	ACIO_CB34_OUT45	$\overline{OUT45}$
ACIO_CB34_OUT36	$\overline{OUT36}$	ACIO_CB34_OUT46	$\overline{OUT46}$
ACIO_CB34_OUT37	$\overline{OUT37}$	ACIO_CB34_OUT47	$\overline{OUT47}$

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

ラッチ機能選択書き込み関数

AL I/O **ユニット対応** DLL

CB-34/IO

機　能

指定された汎用 I/O PORT のラッチ機能を設定します。

書　式

C 言語　BOOL ACIO_BWLatchSel(DWORD *hPort*, WORD FAR **pData*, ACIO_S_RESULT FAR **psResult*);

VB　Function ACIO_BWLatchSel(ByVal *hPort* As Long, *pData* As Integer, *psResult* As ACIO_S_RESULT)　As Boolean

VB.NET　Function ACIO_BWLatchSel(ByVal *hPort* As Integer, ByRef *pData* As Short, ByRef *psResult* As ACIO_S_RESULT) As Boolean

Delphi　function ACIO_BWLatchSel(*hPort*: DWORD; var *pData*: WORD; var *psResult*: ACIO_S_RESULT): Boolean;

引　数

hPort　・・・ $\overline{\text{IN10}}$ ～ $\overline{\text{IN17}}$、$\overline{\text{IN20}}$ ～ $\overline{\text{IN27}}$、$\overline{\text{OUT10}}$ ～ $\overline{\text{OUT17}}$、$\overline{\text{OUT20}}$ ～ $\overline{\text{OUT27}}$ をオープンした汎用 I/O PORT ハンドルを指定します。

pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
変数の内容は、次のようになります。

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	IN13L	IN12L	IN11L	IN10L

IN10L ～ IN13L で、$\overline{\text{IN10}}$ ～ $\overline{\text{IN13}}$ のラッチ機能を選択します。
ラッチ機能を無効にするときに 0、有効にするときに 1 を書き込みます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

ラッチエッジ選択書き込み関数

AL I/O **ユニット対応** DLL

CB-34/IO

機　能

指定された汎用 I/O PORT のラッチのエッジを設定します。
ラッチ機能選択書き込み関数で、ラッチ機能が有効にされているビットのみで有効です。

書　式

C 言語　BOOL ACIO_BWLatchEdge（DWORD *hPort*, WORD FAR **pData*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BWLatchEdge（ByVal *hPort* As Long, *pData* As Integer, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BWLatchEdge（ByVal *hPort* As Integer, ByRef *pData* As Short, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BWLatchEdge（*hPort*: DWORD; var *pData*: WORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

hPort　・・・ $\overline{\text{IN10}}$ ～ $\overline{\text{IN17}}$、$\overline{\text{IN20}}$ ～ $\overline{\text{IN27}}$、$\overline{\text{OUT10}}$ ～ $\overline{\text{OUT17}}$、$\overline{\text{OUT20}}$ ～ $\overline{\text{OUT27}}$ をオープンした汎用 I/O PORT ハンドルを指定します。

pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
変数の内容は、次のようになります。

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	IN13E	IN12E	IN11E	IN10E

IN10E ～ IN13E **で、**$\overline{\text{IN10}}$ ～ $\overline{\text{IN13}}$ **のラッチのエッジを選択します。**
立ち下がりエッジにするときに 0**、立ち上がりエッジにするときに** 1 **を書き込みます。**

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

ラッチ機能選択読み出し関数

AL I/O ユニット対応 DLL

CB-34/IO

機　能

指定された汎用 I/O PORT のラッチ機能の設定を読み出します。

書　式

C 言語　BOOL ACIO_BRLatchSel（DWORD *hPort*, WORD FAR **pData*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BRLatchSel（ByVal *hPort* As Long, *pData* As Integer, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BRLatchSel（ByVal *hPort* As Integer, ByRef *pData* As Short, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BRLatchSel（*hPort*: DWORD; var *pData*: WORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

hPort　・・・ $\overline{\text{IN10}}$ ～ $\overline{\text{IN17}}$、$\overline{\text{IN20}}$ ～ $\overline{\text{IN27}}$、$\overline{\text{OUT10}}$ ～ $\overline{\text{OUT17}}$、$\overline{\text{OUT20}}$ ～ $\overline{\text{OUT27}}$ をオープンした汎用 I/O PORT ハンドルを指定します。

pData　・・・ 読み出した内容が格納されている変数のポインタを指定します。
変数の内容は、次のようになります。

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	IN13L	IN12L	IN11L	IN10L

IN10L ～ IN13L に、$\overline{\text{IN10}}$ ～ $\overline{\text{IN13}}$ のラッチ機能が読み出されます。
ラッチ機能が無効なときに 0、有効なときに 1 が読み出されます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

ラッチエッジ選択読み出し関数

AL I/O ユニット対応 DLL

CB-34/IO

機　能

指定された汎用 I/O PORT のラッチのエッジの設定状態を読み出します。

書　式

C 言語　BOOL ACIO_BRLatchEdge（DWORD *hPort*, WORD FAR **pData*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BRLatchEdge（ByVal *hPort* As Long, *pData* As Integer, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BRLatchEdge（ByVal *hPort* As Integer, ByRef *pData* As Short, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BRLatchEdge（*hPort*: DWORD; var *pData*: WORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

hPort　・・・ $\overline{\text{IN10}}$ ～ $\overline{\text{IN17}}$、$\overline{\text{IN20}}$ ～ $\overline{\text{IN27}}$、$\overline{\text{OUT10}}$ ～ $\overline{\text{OUT17}}$、$\overline{\text{OUT20}}$ ～ $\overline{\text{OUT27}}$ をオープンした汎用 I/O PORT ハンドルを指定します。

pData　・・・ 読み出した内容が格納されている変数のポインタを指定します。
変数の内容は、次のようになります。

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	0	0	0	0	0	IN13E	IN12E	IN11E	IN10E

IN10E ～ IN13E に、$\overline{\text{IN10}}$ ～ $\overline{\text{IN13}}$ のラッチのエッジが読み出されます。
立ち下がりエッジのときに 0、立ち上がりエッジのときに 1 が読み出されます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

ラッチクリア書き込み関数

AL I/O ユニット対応 DLL

CB-34/IO

機　能

指定された汎用 I/O PORT のラッチデータをクリアします。

書　式

C 言語　BOOL ACIO_BWLatchClr（DWORD *hPort*, WORD FAR **pData*, ACIO_S_RESULT FAR **psResult*）;

VB　Function ACIO_BWLatchClr（ByVal *hPort* As Long, *pData* As Integer, *psResult* As ACIO_S_RESULT） As Boolean

VB.NET　Function ACIO_BWLatchClr（ByVal *hPort* As Integer, ByRef *pData* As Short, ByRef *psResult* As ACIO_S_RESULT） As Boolean

Delphi　function ACIO_BWLatchClr（*hPort*: DWORD; var *pData*: WORD; var *psResult*: ACIO_S_RESULT）: Boolean;

引　数

hPort　・・・ $\overline{\text{IN10}}$ ～ $\overline{\text{IN17}}$、$\overline{\text{IN20}}$ ～ $\overline{\text{IN27}}$、$\overline{\text{OUT10}}$ ～ $\overline{\text{OUT17}}$、$\overline{\text{OUT20}}$ ～ $\overline{\text{OUT27}}$ をオープンした汎用 I/O PORT ハンドルを指定します。

pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
変数の内容は、次のようになります。

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	IN13C	IN12C	IN11C	IN10C

IN10C ～ IN13C が、$\overline{\text{IN10}}$ ～ $\overline{\text{IN13}}$ のラッチデータのクリアビットになります。
クリアしないときに 0、クリアするときに 1 を書き込みます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

6. ソフト開発に必要なファイル

ユーザアプリケーション開発に必要なファイルは、インストール時に指定する次のフォルダに格納されています。
(インストール時にパスを\ Program Files 指定した場合)

● AL I/F ユニット対応 DLL を用いたソフト開発に必要なファイル

ファイル	ファイルの場所
Visual C++ヘッダファイル	\ Program Files\ Mpl28\ Bin\ Vc\ AlUsbA.h
Visual C++ライブラリファイル	\ Program Files\ Mpl28\ Bin\ Vc\ VcAlUsbA.lib
C++ Builder ヘッダファイル	\ Program Files\ Mpl28\ Bin\ Builder\ AlUsbA.h
C++ Builder ライブラリファイル	\ Program Files\ Mpl28\ Bin\ Builder\ BcAlUsbA.lib
Visual Basic .NET 関数定義ファイル	\ Program Files\ Mpl28\ Bin\ Vb.NET\ AlUsbA.vb
Visual Basic 関数定義ファイル	\ Program Files\ Mpl28\ Bin\ Vb\ AlUsbA.bas

● AL I/O ユニット対応 DLL を用いたソフト開発に必要なファイル

ファイル	ファイルの場所
Visual C++ヘッダファイル	\ Program Files\ Mpl28\ Bin\ Vc\ AcIoA.h
Visual C++ライブラリファイル	\ Program Files\ Mpl28\ Bin\ Vc\ VcAcIoA.lib
C++ Builder ヘッダファイル	\ Program Files\ Mpl28\ Bin\ Builder\ AcIoA.h
C++ Builder ライブラリファイル	\ Program Files\ Mpl28\ Bin\ Builder\ BcAcIoA.lib
Visual Basic .NET 関数定義ファイル	\ Program Files\ Mpl28\ Bin\ Vb.NET\ AcIoA.vb
Visual Basic 関数定義ファイル	\ Program Files\ Mpl28\ Bin\ Vb\ AcIoA.bas

● AL MCC05 ユニット対応 DLL を用いたソフト開発に必要なファイル
AL MCC05 ユニット取扱説明書をご覧ください

● AL MCC06 ユニット対応 DLL を用いたソフト開発に必要なファイル
AL MCC06 ユニット取扱説明書をご覧ください

7. サンプルプログラム

7-1. 概要

Visual C++ .NET、Visual C++、C++ Builder、Visual Basic .NET、Visual Basic のサンプルプログラムが用意されています。
サンプルプログラムのファイルは、インストール時に指定する次のフォルダに格納されています。
（インストール時にパスを Program Files と指定した場合）

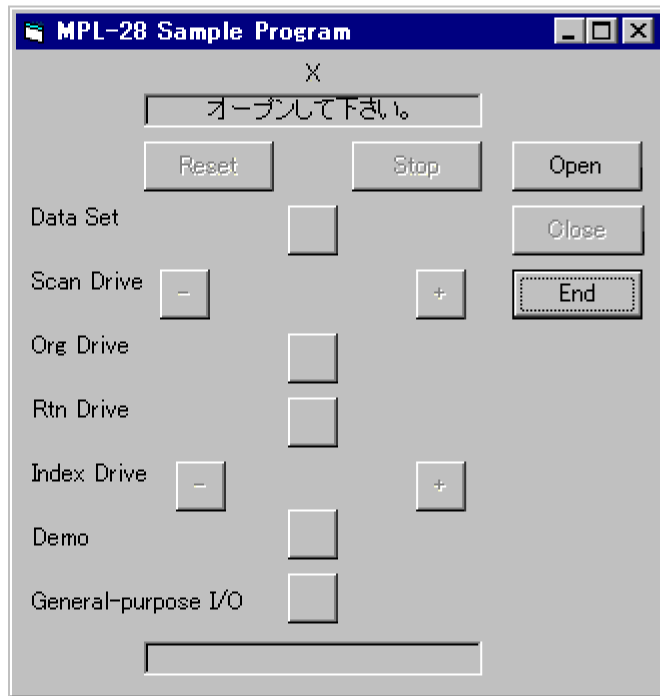
言語	ファイルの場所
Visual C++ .NET、Visual C++	\ Program Files\ Mpl28\ Sample\ Vc
C++ Builder	\ Program Files\ Mpl28\ Sample\ Builder
Visual Basic .NET	\ Program Files\ Mpl28\ Sample\ Vb.NET
Visual Basic	\ Program Files\ Mpl28\ Sample\ Vb

7-2. ご使用になる前に

環境設定ツールで AL 通信ボーレート、リトライ回数を設定して下さい。
パソコン、CB-23/USB、C-772（SLAVE ADDRESS = H'01）、CB-08（SLAVE ADDRESS = H'02）を接続して下さい。

7-3.　仕様

サンプルプログラムには、Visual C++ 6.0、C++ Builder 5.0、Visual Basic .NET 2002、Visual Basic 6.0 で作成したものを用意してあります。これらは、同じ仕様で作られています。
サンプルプログラムを参照する場合には、それぞれの言語の開発環境からプロジェクトを開いて下さい。



Open ボタン	デバイス(C-772 X 軸)をオープンします。
Close ボタン	デバイス(C-772 X 軸)をクローズします。
End ボタン	サンプルプログラムを終了します。
Reset ボタン	ADDRESS COUNTER を 0 に PRESET します。
Stop ボタン	DRIVE を即時停止します。
Data Set ボタン	次の設定にします。 RATE TYPE : L1-TYPE URATE : 10ms/1000Hz　LSPD : 1000Hz DRATE : 10ms/1000Hz　HSPD : 5000Hz
Scan Drive +ボタン	+(CW)方向へ SCAN DRIVE します。
Scan Drive -ボタン	-(CCW)方向へ SCAN DRIVE します。
Org Drive ボタン	機械原点検出形式 ORG-3 で ORIGIN DRIVE を行います。
Rtn Drive ボタン	絶対 ADDRESS 0 へ移動する ABSOLUTE INDEX DRIVE を行います。
Index Drive +ボタン	+(CW)方向へ 3000 パルス移動する INCREMENTAL INDEX DRIVE を行います。
Index Drive -ボタン	-(CCW)方向へ 3000 パルス移動する INCREMENTAL INDEX DRIVE を行います。
Demo ボタン	次の動作を連続して行います。 機械原点の検出(ORIGIN DRIVE) 電気原点の設定(ADDRESS COUNTER を 0 に PRESET) +(CW)方向へ 4000 パルス移動を 4 回繰り返す(INCREMENTAL INDEX DRIVE) 絶対 ADDRESS 30000 へ移動(ABSOLUTE INDEX DRIVE) 電気原点へ移動(ABSOLUTE INDEX DRIVE)
General-Purpose I/O ボタン	次の動作を連続して行います。 I/O PORT OPEN(CB-08) IN10 - IN12 が HIGH になるまで待つ OUT10 - OUT12 を HIGH にする +(CW)方向へ 1000 パルス移動(INCREMENTAL INDEX DRIVE) OUT10 を LOW にする IN11 が LOW になるまで待つ +(CW)方向へ 2000 パルス移動(INCREMENTAL INDEX DRIVE) OUT11 を LOW にする IN12 が LOW になるまで待つ +(CW)方向へ 3000 パルス移動(INCREMENTAL INDEX DRIVE) OUT12 を LOW にする I/O PORT CLOSE(CB-08)

8.　トラブルシューティング

作成したアプリケーションプログラムが正常に動作しない場合、次のことを行って下さい。

(1) I/F ユニットの認識
スレーブ情報確認プログラムを起動して下さい。
スレーブ情報確認プログラムは、起動時に I/F ユニットの認識を確認しています。

(2) スレーブの認識
スレーブ情報確認プログラムは、実際に USB 通信、AL 通信を行い、各スレーブアドレスに接続されているスレーブを検出し、検出結果を表示しています。

メッセージ	スレーブの検出結果
Unconnection	スレーブが接続されていないことを示しています。
C-770AL	C-770AL が接続されていることを示しています。
CB-08	CB-08 が接続されていることを示しています。
CDB-5420-AL770	CDB-5420-AL770 が接続されていることを示しています。
CAD-5410-AL770	CAD-5410-AL770 が接続されていることを示しています。
C-771	C-771 が接続されていることを示しています。
CD-773/ADB5331A	CD-773/ADB5331A が接続されていることを示しています。
C-772	C-772 が接続されていることを示しています。
CB-34	CB-34 が接続されていることを示しています。

(3) RESULT 構造体の確認
各関数は、アプリケーションプログラムによって与えられた引数の内容をチェックし、エラーがある場合は、FALSE(0)を返し、正常である場合は TRUE(1)を返します。
関数が正常に動作していないと思われるステップの後にブレークポイントを設定し、関数が返した値が TRUE(1)であることを確認して下さい。
TRUE(1)でない場合は、エラー原因を特定する為に RESULT 構造体の内容を参照して下さい。

本版で改訂された主な箇所

箇　所	内　容
なし	

■ 製品保証

保証期間と保証範囲について

- 納入品の保証期間は、納入後１ヶ年と致します。
- 上記保証期間中に当社の責により故障を生じた場合は、その修理を当社の責任において行います。
(日本国内のみ)
ただし、次に該当する場合は、この保証対象範囲から除外させて頂きます。
(1) お客様の不適当な取り扱い、ならびに使用による場合。
(2) 故障の原因が、当製品以外からの事由による場合。
(3) お客さまの改造、修理による場合。
(4) 製品出荷当時の科学・技術水準では予見が不可能だった事由による場合。
(5) その他、天災、災害等、当社の責にない場合。
(注1)ここでいう保証は、納入品単体の保証を意味するもので、納入品の故障により誘発される損害はご容赦頂きます。
(注2)当社において修理済みの製品に関しましては、保証外とさせて頂きます。

技術相談のお問い合わせ

TEL.(042)664-5382 FAX.(042)666-5664
E-mail s-support@melec-inc.com

販売に関するお問い合わせ

TEL.(042)664-5384 FAX.(042)666-2031

株式会社メレック 制御機器営業部
〒193-0834 東京都八王子市東浅川町516-10
URL:http://www.melec-inc.com

記載内容は、製品改良のため予告なく変更することがありますのでご了承ください。